



## JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11)Publication number: 07281890

(43)Date of publication of application: 27.10.1995

(51)Int.Cl.

G06F 9/305  
G06F 9/30(21)Application number: 06068731 (71)Applicant: MITSUBISHI ELECTRIC CORP  
(22)Date of filing: 06.04.1994 (72)Inventor: ITO SAKAE

(54) INSTRUCTION SET AND ITS EXECUTING METHOD BY MICROCOMPUTER

(57)Abstract:

**PURPOSE:** To provide an instruction set and its executing method by microcomputer which increase the number of usable instructions even when the operand size of the operation object of each instruction is plural, and lighten the load at the time of programming.

**CONSTITUTION:** The instruction set consisting of a 1st instruction subset IS1 composed of plural instructions which perform specific operation for an operand of certain size while codes are assigned uniquely and respectively and a 2nd instruction subset IS2 composed of plural instructions to which the codes of the instructions of the 1st instruction subset IS1 performing the same operation are assigned as low-order codes and a common prefix code is added as the high-order codes, is so executed to specify operand sizes that the instruction subsets IS1 and IS2 handle according to

ニーニツク	IS1	IS2
	第1命令サブセット 命令(12進表示)	第2命令サブセット 命令コード(2進表示)
ADD	00000000	11111111 00000000
MOV	00000001	11111111 00000001
AND	00000010	11111111 00000010
...	...	...

命令コードの11111111は第1命令サブセットの  
命令コードの2進表示である。

the contents of a flag register.

---

#### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

---

Copyright (C); 1998 Japanese Patent Office

---

[MENU](#)

[SEARCH](#)

[INDEX](#)

[DETAIL](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-281890

(43) 公開日 平成7年(1995)10月27日

(51) Int. Cl. <sup>4</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/305				
9/30	3 5 0 A		G 0 6 F 9/ 30	3 4 0 E

審査請求 未請求 請求項の数 6 O L (全 12 頁)

(21) 出願番号 特願平6-68731

(22) 出願日 平成6年(1994)4月6日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 伊藤 栄

兵庫県伊丹市瑞原4丁目1番地 三菱電機

株式会社北伊丹製作所内

(74) 代理人 弁理士 河野 登夫

(54) 【発明の名称】 命令セット及びそのマイクロコンピュータによる実行方法

(57) 【要約】

【目的】 各命令の操作対象のオペランドサイズが複数である場合にも、使用可能な命令の数を可能な限り多くし、またプログラミングの際の負担を軽減し得る命令セット及びそのマイクロコンピュータによる実行方法を提供する。

【構成】 それぞれに一意にコードが割り付けられており、あるサイズのオペランドに対してそれぞれ所定の操作を実行する複数の命令にて構成された第1命令サブセット IS1 と、それぞれに第1命令サブセット IS1 の同一の操作を実行する命令のコードが下位コードとして割り付けられており、上位コードとしてそれぞれに共通の接頭コードが付加された複数の命令にて構成された第2命令サブセット IS2 とで構成された命令セットをフラグレジスタの内容に応じて各命令サブセット IS1、IS2 が扱うオペランドサイズを指定するように実行する。

ニーモニック	IS1		IS2	
	第1命令サブセット 命令コード(2進表示)		第2命令サブセット 命令コード(2進表示)	
ADD	00000000		11111111	00000000
SBB	00000001		11111111	00000001
AND	00000010		11111111	00000010
⋮	⋮		⋮	⋮

\*ただし、“11111111”は第1命令サブセットの命令コードとしては使用出来ない

## 【特許請求の範囲】

【請求項1】 複数のサイズのオペランドを操作する複数の命令にて構成されたマイクロコンピュータの命令セットであって、

前記複数のサイズの内のいずれかのサイズのオペランドに対してそれぞれ所定の操作を実行すると共に、それぞれ異なるコードが割り付けられた複数の命令にて構成された基本命令セットと、

前記複数のサイズの内のいずれかのサイズのオペランドに対してそれぞれ所定の操作を実行すると共に、それぞれに下位コードとして前記基本命令セットの同一の操作を実行する命令のコードが割り付けられており、上位コードとしてそれぞれに共通の接頭コードが付加された複数の命令にてそれぞれが構成された第2の命令セットを含む複数の命令セットとで構成されたことを特徴とするマイクロコンピュータの命令セット。

【請求項2】 第1または第2のサイズのオペランドを操作する複数の命令にて構成されたマイクロコンピュータの命令セットであって、

第1または第2のサイズのオペランドに対してそれぞれ所定の操作を実行すると共に、それぞれ異なるコードが割り付けられた複数の命令にて構成された第1の命令セットと、

第1または第2のサイズのオペランドに対してそれぞれ所定の操作を実行すると共に、それぞれに下位コードとして前記第1の命令セットの同一の操作を実行する命令のコードが割り付けられており、上位コードとして共通の接頭コードが付加された複数の命令にて構成された第2の命令セットとで構成されたことを特徴とするマイクロコンピュータの命令セット。

【請求項3】 第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間で演算を実行して演算結果を前記第1または前記第2のデータ格納場所に格納する複数の命令にて構成されたマイクロコンピュータの命令セットであって、

第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間でそれぞれ所定の演算を実行して演算結果を前記第1または第2のデータ格納場所に格納すると共に、それぞれ異なるコードが割り付けられた複数の命令にて構成された第1の命令セットと、

第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間でそれぞれ所定の演算を実行して演算結果を前記第1または第2のデータ格納場所に格納すると共に、それぞれに下位コードとして前記第1の命令セットの同一の演算を実行する命令のコードが割り付けられており、上位コードとして共通の接頭コードが付加された複数の命令にて構成された第2の命令セットとで構成されたことを特徴とするマイクロコンピュータの命令セット。

【請求項4】 その内容を任意に設定可能なフラグレジスタを備え、複数のサイズのオペランドを操作する複数の命令にて構成された命令セットのマイクロコンピュータによる実行方法であって、

それぞれに割り付けられた異なるコードを有し、前記複数のサイズの内のいずれかのサイズのオペランドに対してそれぞれ所定の操作を実行する複数の命令にて構成された基本命令セットと、

それぞれに下位コード前記基本命令セットの各命令のコードがとして割り付けられると共に上位コードとして共通の接頭コードが付加されたコードを有し、付加されている接頭コードの内容に対応して前記複数のサイズの内のいずれかのサイズのオペランドに対して下位コードが同一の前記基本命令セットの命令と同一の操作を実行する複数の命令にてそれぞれが構成された複数の命令セットとを、前記各命令セットが、前記フラグレジスタの内容と、接頭コードの有無、または付加されている接頭コードの内容とに応じてそれぞれ異なるサイズのオペランドを操作するようにしたことを特徴とする命令セットのマイクロコンピュータによる実行方法。

【請求項5】 その内容を任意に設定可能な1ビットのフラグレジスタを備え、第1のサイズまたは第2のサイズのオペランドを操作する複数の命令にて構成された命令セットのマイクロコンピュータによる実行方法であって、

それぞれに割り付けられた異なるコードを有し、第1または第2のサイズのオペランドに対してそれぞれ所定の操作を実行する複数の命令にて構成された第1の命令セットと、

それぞれに下位コードとして前記第1の命令セットの各命令のコードが割り付けられると共に上位コードとして共通の接頭コードが付加されたコードを有し、第1または第2のサイズのオペランドに対して下位コードが同一の前記第1の命令セットの命令と同一の操作を実行する複数の命令にて構成された第2の命令セットとを、前記フラグレジスタの内容が第1の状態にある場合は、前記第1の命令セットの各命令が第1のサイズのオペランドを操作し、前記第2の命令セットの各命令が第2のサイズのオペランドを操作するように、

また、前記フラグレジスタの内容が第2の状態にある場合は、前記第1の命令セットの各命令が第2のサイズのオペランドを操作し、前記第2の命令セットの各命令が第1のサイズのオペランドを操作するようにしたことを特徴とする命令セットのマイクロコンピュータによる実行方法。

【請求項6】 その内容を任意に設定可能な1ビットのフラグレジスタを備え、第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間で演算を実行して演算結果を前記第1または前記第2のデータ格納場所に格納する複数の命令にて構成さ

れた命令セットのマイクロコンピュータによる実行方法であって、

それぞれに割り付けられた異なるコードを有し、第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間でそれぞれ所定の演算を実行して演算結果を前記第1または第2のデータ格納場所に格納する複数の命令にて構成された第1の命令セットと、

それぞれに下位コードとして前記第1の命令セットの各命令のコードが割り付けられると共に上位コードとして共通の接頭コードが付加されたコードを有し、第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間で下位コードが同一の前記第1の命令セットの命令と同一の演算を実行して演算結果を前記第1または第2のデータ格納場所に格納する複数の命令にて構成された第2の命令セットとを、

前記フラグレジスタの内容が第1の状態にある場合は、前記第1の命令セットの各命令は演算結果を前記第1の格納場所に格納し、前記第2の命令セットの各命令は演算結果を前記第2の格納場所に格納するように、

また、前記フラグレジスタの内容が第2の状態にある場合は、前記第1の命令セットの各命令は演算結果を前記第2の格納場所に格納し、前記第2の命令セットの各命令は演算結果を前記第1の格納場所に格納するようにしたことを特徴とする命令セットのマイクロコンピュータによる実行方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はマイクロコンピュータの命令セットに関し、またそのマイクロコンピュータによる実行方法に関する。

【0002】

【従来の技術】従来の技術の一例として、8ビット長と16ビット長との二つのオペランドサイズを処理することが可能な命令セット及びそのマイクロコンピュータによる実行方法について以下に説明する。たとえば、マイクロコンピュータに記憶された命令によりレジスタの内容とメモリの内容とを加算してその結果を元のレジスタに格納する場合のデータの移動状態を図4及び図5の模式図に示す。なお、通常のマイクロコンピュータではメモリアドレスは8ビット単位で割り付けられているが、16ビット長のオペランドを処理可能な場合にはレジスタは16ビット幅を有する。

【0003】オペランドサイズが8ビットである場合には、図4に示されているように、第1のソースオペランドとして16ビット長のレジスタRG1の下位側の8ビットの内容が被加数に、第2のソースオペランドとしてメモリMの指定されたメモリアドレスに格納されている8ビットのデータが加数にそれぞれなって両者が加算され、デスティネーションオペランドとしての8ビットの加算

結果はレジスタRG1の下位側の8ビット、即ち被加数の元の格納場所に格納される。

【0004】一方、オペランドサイズが16ビットである場合には、図5に示されているように、16ビット長のレジスタRG2の全体の内容が被加数に、メモリMの指定されたメモリアドレスとそれに連続するもう一つのメモリアドレスとの双方に格納されている合計16ビットのデータが加数にそれぞれなって両者が加算され、16ビットの加算結果はレジスタRG2の全体、即ち被加数の元の格納場所に格納される。

【0005】このように、同一の操作であっても異なるオペランドサイズを処理する場合、たとえば上述のようなオペランドサイズが8ビット長または16ビット長のいずれかを処理する場合には、それぞれに対応した命令が必要になる。そして、そのような命令が複数、マイクロコンピュータにより実行される場合には以下の二通りの手法が考えられる。

【0006】その第1は、8ビット長のオペランドのみを操作する命令と、16ビット長のオペランドのみを操作する命令とをそれぞれ全く異なる命令としてそれぞれに異なる命令コード、換言すれば一意に命令コードを割り当てる手法である。

【0007】具体的には、図6の命令セットの模式図に示されているように、たとえば加算命令では、8ビットオペランド用には命令コードとして"00000000"が割り当てられたニーモニック"ADD.B"で示されている専用の加算命令が用意され、16ビットオペランド用には命令コードとして"00000001"が割り当てられたニーモニック"ADD.W"で示されている専用の加算命令が用意される。

【0008】なお、ニーモニックの末尾の"B"はオペランドがバイトサイズ、即ち8ビット長であることを、また"W"はオペランドがワードサイズ、即ち16ビット長であることをそれぞれ示している。

【0009】また、減算命令では、8ビットオペランド用には命令コードとして"00000010"が割り当てられたニーモニック"SUB.B"で示されている専用の減算命令が用意され、16ビットオペランド用には命令コードとして"00000011"が割り当てられたニーモニック"SUB.W"で示されている専用の減算命令が用意される。

【0010】この第1の手法では、オペランドサイズは異なるものの、基本的には全く同一の操作内容の命令に対して二種類の命令コードを割り当てる必要がある。従って、各命令コード長をたとえば8ビット長とした場合には、最大で256個使用可能な命令コードが8ビット長用と16ビット長用とに二分されて実質的には128種類の命令が使用可能になる。但し、命令コードを16ビット長とした場合には、命令コードが8ビット長である場合に比して更に多数の命令コードが使用可能になる。しかし、そのように命令コード長を長くした場合には、プロ

グラムサイズ、即ちプログラムのデータ量が大きくなってそれを格納するためのマイクロコンピュータのメモリも大容量が必要になり、また処理速度が遅くなる等の新たな問題が生じる。

【0011】第2の手法は、一つの操作内容に対して命令コードを一つのみ割り当て、オペランドサイズが8ビットであるかまたは16ビットであるかは1ビットのフラグレジスタにより指示する手法である。この場合、たとえばフラグレジスタの内容が“1”であればいずれの命令を実行した場合にもオペランドサイズとしては8ビット長が指定され、フラグレジスタの内容が“0”であればいずれの命令を実行した場合にもオペランドサイズとしては16ビット長が指定されるようにする。

【0012】具体的には、図7の模式図に示されているように、たとえば加算命令では、命令コードとして“00000000”が割り当てられたニーモニック“ADD”で示されている加算命令のみが用意され、8ビットオペランドを処理する場合にはフラグレジスタの内容が“1”に設定され、16ビットオペランドを処理する場合にはフラグレジスタの内容が“0”に設定される。また、減算命令では、命令コードとして“00000001”が割り当てられたニーモニック“SUB”で示されている減算命令のみが用意され、8ビットオペランドを処理する場合にはフラグレジスタの内容が“1”に設定され、16ビットオペランドを処理する場合にはフラグレジスタの内容が“0”に設定される。

【0013】このような第2の手法では、命令コード長が8ビットであっても実質的に256種類の命令が使用可能になるが、フラグレジスタを“1”または“0”にセット、リセットする命令が別途必要になる。また、プログラミングに際しては、プログラマはフラグレジスタがいずれの値になっているかということを常時念頭に置いておかなければ正しいプログラムを作成することが出来ない等の問題が生じる。

【0014】次に、従来の技術の他の例としてたとえば、レジスタの内容とメモリの内容とにある演算、たとえば加算を行なってその結果を元のレジスタ、またはメモリに格納する場合のデータの移動状態を図8及び図9の模式図に示す。

【0015】演算結果がレジスタに格納される場合には図8に示されているように、第1のソースオペランドとして8ビット長のレジスタRG3の内容が被加数に、第2のソースオペランドとしてメモリMの指定されたメモリアドレスに格納されている8ビットのデータが加数にそれぞれなって両者が加算され、デスティネーションオペランドとしての8ビットの加算結果はレジスタRG3、即ち被加数の元の格納場所（第1のソースオペランド）に格納される。

【0016】一方、演算結果がメモリに格納される場合には、図9に示されているように、第1のソースオペ

ランドとしてレジスタRG3の内容が被加数に、第2のソースオペランドとしてメモリMの指定されたメモリアドレスに格納されている8ビットのデータが加数にそれぞれなって両者が加算され、デスティネーションオペランドとしての8ビットの加算結果はメモリM、即ち加数の元の格納場所（第2のソースオペランド）に格納される。

【0017】このように、同一の演算であっても演算結果の格納先、即ちデスティネーションが異なる処理を行なう場合、たとえば上述のようなデスティネーションが第1のソースオペランドとしてのレジスタである場合と第2のソースオペランドとしてのメモリである場合とのいずれかの処理を行なう場合には、それぞれに対応した命令が必要になる。そして、そのような命令が複数、マイクロコンピュータにより実行される場合には以下の二通りの手法が考えられる。

【0018】その第1は、デスティネーションオペランドが第1のソースオペランドである命令と、デスティネーションオペランドが第2のソースオペランドである命令とをそれぞれ全く異なる命令としてそれぞれに一意に命令コードを割り当てる手法である。

【0019】具体的には、図10の模式図に示されているように、たとえば加算命令では、デスティネーションが第1ソース（レジスタ）である命令用には命令コードとして“00000000”が割り当てられたニーモニック“ADD.1”で示されている専用の加算命令が用意され、デスティネーションが第2ソース（メモリ）である命令用には命令コードとして“00000001”が割り当てられたニーモニック“ADD.2”で示されている専用の加算命令が用意される。

【0020】なお、ニーモニックの末尾の“1”はデスティネーションが第1ソース、即ちレジスタであることを、また“2”はデスティネーションが第2ソース、即ちメモリであることをそれぞれ示している。

【0021】また、減算命令では、デスティネーションが第1ソースである命令用には命令コードとして“00000010”が割り当てられたニーモニック“SUB.1”で示されている専用の減算命令が用意され、デスティネーションが第2ソースである命令用には命令コードとして“00000011”が割り当てられたニーモニック“SUB.2”で示されている専用の減算命令が用意される。

【0022】この第1の手法では、デスティネーションは異なるものの基本的には全く同一の操作内容の命令に対して二種類の命令コードを割り当てる必要があり、前述のオペランドサイズが異なる場合の第1の手法と同様の問題が生じる。

【0023】第2の手法は、一つの操作内容に対して命令コードを一つのみ割り当て、デスティネーションが第1ソースであるかまたは第2ソースであるかは1ビットのフラグレジスタにより指示する手法である。この場合、たとえばフラグレジスタの内容が“1”であればオ

ペラントソースが二つ有るいずれの命令を実行した場合にもデスティネーションとしては第1ソースが指定され、フラグレジスタの内容が“0”であればいずれの命令を実行した場合にもデスティネーションとしては第2ソースが指定されるようにする。

【0024】具体的には、図11の模式図に示されているように、たとえば加算命令では、命令コードとして“00000000”が割り当てられたニーモニック“ADD”で示されている加算命令のみが用意され、デスティネーションが第1ソースである命令を処理する場合にはフラグレジスタの内容が“1”に設定され、デスティネーションが第2ソースである命令を処理する場合にはフラグレジスタの内容が“0”に設定される。また、減算命令では、命令コードとして“00000001”が割り当てられたニーモニック“SUB”で示されている減算命令のみが用意され、デスティネーションが第1ソースである命令を処理する場合にはフラグレジスタの内容が“1”に設定され、デスティネーションが第2ソースである命令を処理する場合にはフラグレジスタの内容が“0”に設定される。

【0025】このような第2の手法では、命令コード長が8ビットであっても実質的に256種類の命令が使用可能になるが、前述のオペランドサイズが異なる場合の第2の手法と同様の問題が生じる。

【0026】

【発明が解決しようとする課題】このように、従来のマイクロコンピュータの命令セットでは、実質的に同一の操作、演算を実行する際に、オペランドサイズが複数である場合、あるいは演算結果の格納場所が複数である場合にはそれぞれに対応した命令を用意するか、またはフラグレジスタの設定値によりいずれかを選択するようにしていた。このため、前者では設定可能な命令の数に比べて使用可能な命令の数が実質的には1/2あるいはそれ以下になり、後者ではプログラミングに際してプログラマの負担が大きいという問題があった。

【0027】本発明はこのような事情に鑑みてなされたものであり、各命令の操作対象のオペランドサイズが複数である場合にも、また演算結果の格納場所が複数である場合にも、使用可能な命令の数を設定可能な命令の数に実質的に等しくすることを可能とし、またプログラミングの際のプログラマの負担を軽減し得る命令セット及びそのマイクロコンピュータによる実行方法の提供を目的とする。

【0028】

【課題を解決するための手段】本発明の命令セットの第1の発明は、それぞれに一意にコードが割り付けられており、あるサイズのオペランドに対してそれぞれ所定の操作を実行する複数の命令にて構成された基本命令セットと、それぞれに基本命令セットの同一の操作を実行する命令のコードが下位コードとして割り付けられてお

り、上位コードとしてそれぞれに共通の接頭コードが付加された複数の命令にて構成された第2の命令セットを含む複数の命令セットとで構成されている。

【0028】本発明の命令セットの第2の発明は、それぞれに一意にコードが割り付けられており、あるサイズのオペランドに対してそれぞれ所定の操作を実行する複数の命令にて構成された第1の命令セットと、それぞれに第1の命令セットの同一の操作を実行する命令のコードが下位コードとして割り付けられており、上位コードとしてそれぞれに共通の接頭コードが付加された複数の命令にて構成された第2の命令セットとで構成されている。

【0030】本発明の命令セットの第3の発明は、それぞれに一意にコードが割り付けられており、第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間でそれぞれ所定の演算を実行して演算結果を第1または第2のデータ格納場所に格納する複数の命令にて構成された第1の命令セットと、それぞれに第1の命令セットの同一の操作を実行する命令のコードが下位コードとして割り付けられており、上位コードとしてそれぞれに共通の接頭コードが付加された複数の命令にて構成された第2の命令セットとで構成されている。

【0031】本発明の命令セットのマイクロコンピュータによる実行方法の第1の発明は、その内容を任意に設定可能なフラグレジスタを備え、複数のサイズのオペランドを操作する複数の命令を実行する場合に、前述の命令セットの第1の発明の各命令セットをそれぞれが、フラグレジスタの内容と、接頭コードの有無、または付加されている接頭コードの内容とに応じてそれぞれ異なるサイズのオペランドを操作するように実行する。

【0032】本発明の命令セットのマイクロコンピュータによる実行方法の第2の発明は、その内容を任意に設定可能な1ビットのフラグレジスタを備え、複数のサイズのオペランドを操作する複数の命令を実行する場合に、前述の命令セットの第2の発明の第1及び第2の命令セットを、フラグレジスタの内容が第1の状態にある場合は、第1の命令セットの各命令が第1のサイズのオペランドを操作し、第2の命令セットの各命令が第2のサイズのオペランドを操作するように、またフラグレジスタの内容が第2の状態にある場合は、第1の命令セットの各命令が第2のサイズのオペランドを操作し、第2の命令セットの各命令が第1のサイズのオペランドを操作するように実行する。

【0033】本発明の命令セットのマイクロコンピュータによる実行方法の第3の発明は、その内容を任意に設定可能な1ビットのフラグレジスタを備え、第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間で演算を実行して演算結果を第1または第2のデータ格納場所に格納する複数の命令

を実行する場合に、前述の命令セットの第2の発明の第1及び第2の命令セットを、フラグレジスタの内容が第1の状態にある場合は、第1の命令セットの各命令は演算結果を第1の格納場所に格納し、第2の命令セットの各命令は演算結果を第2の格納場所に格納するように、またフラグレジスタの内容が第2の状態にある場合は、第1の命令セットの各命令は演算結果を第2の格納場所に格納し、第2の命令セットの各命令は演算結果を第1の格納場所に格納するように実行する。

【0034】

【作用】本発明の命令セットの第1の発明では、基本命令セット以外の複数の命令セットのそれぞれにおいては、基本命令セットが対象とするサイズとはそれぞれ異なるサイズのオペランドに対して基本命令セットの各命令のコードと同一の下位コードを有している命令が同一の操作を実行する。

【0035】本発明の命令セットの第2の発明では、第2の命令セットにおいては、第1の命令セットが対象とするサイズとは異なるサイズのオペランドに対して第1の命令セットの各命令のコードと同一の下位コードを有している命令が同一の操作を実行する。

【0036】本発明の命令セットの第3の発明では、第2の命令セットにおいては、第1の命令セットの各命令のコードと同一の下位コードを有している命令が同一の演算を実行し、第1の命令セットの各命令が演算結果を第1のデータ格納場所に格納するのであれば第2の格納場所に、第1の命令セットの各命令が演算結果を第2のデータ格納場所に格納するのであれば第1の格納場所に格納する。

【0037】本発明の命令セットのマイクロコンピュータによる実行方法の第1の発明では、フラグレジスタの内容と、接頭コードの有無、または付加されている接頭コードの内容とに応じて各命令セットの下位コードが同一の命令が異なるサイズのオペランドに対して同一の操作を実行する。

【0038】本発明の命令セットのマイクロコンピュータによる実行方法の第2の発明では、フラグレジスタの内容と接頭コードの内容とに応じて第1と第2の命令セットの下位コードが同一の命令が異なるサイズのオペランドに対して同一の操作を実行する。

【0039】本発明の命令セットのマイクロコンピュータによる実行方法の第3の発明では、フラグレジスタの内容と接頭コードの内容とに応じて第1と第2の命令セットの下位コードが同一の命令が第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間で同一の演算を実行して演算結果をそれぞれ異なるデータ格納場所に格納する。

【0040】

【実施例】以下、本発明をその実施例を示す図面に基

り付けられている命令コードを示す模式図である。

【0041】本発明の命令セットはたとえば8ビットオペランドと16ビットオペランドとの二種類のサイズのオペランドを処理する場合には二組の命令サブセットが用意される。具体的には、本実施例では、図1において参照符号IS1にて示されている基本命令セットとしての第1命令サブセットと、参照符号IS2で示されている第2命令サブセットとが用意される。

【0042】第1命令サブセットIS1は、一つの操作に対して一つの命令コードが割り当てられている。この第1命令サブセットIS1では、命令コードは8ビット長であるので、最大で256個(種類)の命令をマイクロコンピュータに記憶させて実行させることが可能になる。

【0043】また、第2命令サブセットIS2の各命令コードは第1命令サブセットIS1の各命令コードに8ビット長の適当な接頭語、たとえば“1111111”を付加して構成されている。従って、第2命令サブセットIS2の各命令コードは16ビット長になる。但し、第2命令サブセットIS2の接頭語として使用されるコードは第1命令サブセットIS1の命令コードとしては使用できない。具体的には、本実施例では第2命令サブセットIS2の接頭語として“1111111”が使用されるため、第1命令サブセットIS1の命令コードとして“1111111”を使用することはできない。

【0044】更に、上述のような命令セットのマイクロコンピュータによる実行方法の第1の実施例として、1ビットのフラグレジスタが使用される。そして、図2に示されているように、フラグレジスタの内容が“0”である場合には、第1命令サブセットIS1の各命令のオペランドサイズは8ビット長として処理され、また同時に第2命令サブセットIS2の各命令のオペランドサイズは16ビット長として処理される。

【0045】一方、フラグレジスタの内容が“1”である場合には、上述とは逆に、第1命令サブセットIS1の各命令のオペランドサイズは16ビット長として処理され、また同時に第2命令サブセットIS2の各命令のオペランドサイズは8ビット長として処理される。

【0046】従って、プログラマは、自身が作成するプログラムで扱われるオペランドのサイズが主として8ビットである場合にはフラグレジスタの内容が“0”に設定されるようにたとえばプログラムの最初にプログラミングする。そして、それ以降は、主として各命令コードが8ビットで構成される第1命令サブセットIS1を使用してプログラムを作成し、16ビット長のオペランドサイズを扱う命令が必要な場合にのみ、第2命令サブセットIS2の命令を使用してプログラミングを行なう。

【0047】一方、プログラマは、プログラムで扱われるオペランドのサイズが主として16ビットである場合にはフラグレジスタの内容が“1”に設定されるようにたとえばプログラムの最初にプログラミングする。そし



## 11

て、それ以降は、主として各命令コードが8ビットで構成される第1命令サブセットIS1を使用してプログラムを作成し、8ビット長のオペランドサイズを扱う命令が必要な場合のみ、第2命令サブセットIS2の命令を使用してプログラミングを行なう。

【0048】このように、プログラム中で扱われるオペランドのサイズが主として8ビットである場合にはフラグレジスタを"0"に、16ビットである場合にはフラグレジスタを"1"にそれぞれセットすることにより、いずれの場合にも8ビット長の命令コードで構成される第1命令サブセットIS1を主として使用してプログラムを行なうことが可能になる。従って、第2命令サブセットIS2の16ビット長の命令コードは全く使用されないか、あるいは使用頻度が非常に低くなるので、プログラムサイズを小さくすることが可能になり、また命令の実行速度も高速化される。更に、フラグレジスタの内容をプログラムの途中で変更する必要もない。

【0049】なお、上述の本発明の命令セットのマイクロコンピュータによる実行方法の第1の実施例では、命令セットは基本命令セットとしての第1命令サブセットIS1と、この第1命令サブセットIS1の各命令の8ビットのコードに8ビットのを接頭コード"11111111"が付加された16ビットのコードを有する命令にて構成される第2命令サブセットIS2とで構成されているが、各命令セットに共通の下位コード長、換言すれば第1命令サブセットIS1のコード長は8ビットに限定されるものではない。また、接頭コードは"11111111"に限定されるものではなく、更に8ビットに限定されるものでもないが、基本の命令セットのコード長と同一ビット数とすることが望ましい。

【0050】また、第1命令サブセットIS1の各命令のコードに異なる接頭コードを付加することにより、第1命令サブセットIS1及び第2命令サブセットIS2の他に更にいくつかの命令サブセットを備え、それぞれの命令サブセットに異なるサイズのオペランドを操作させることも勿論可能である。なお、そのような場合にも本発明の趣旨からすれば、基本となる第1命令サブセットIS1の各命令のコードはあくまでも8ビットで構成し、最も使用頻度が高いオペランドサイズを第1命令サブセットIS1で操作することが望ましい。

【0051】次に、本発明の命令セットのマイクロコンピュータによる実行方法の第2の実施例について図面に基づいて説明する。

【0052】ここでは、図1に示されている8ビットの命令コードからなる第1命令サブセットIS1と16ビットの命令コードからなる第2命令サブセットIS2とのマイクロコンピュータによる実行方法の第2の実施例として図3に示されているように、フラグレジスタの内容が"0"である場合には、第1命令サブセットIS1の各命令のデスティネーションオペランド、即ち演算結果の格納

## 12

先が第1ソースとして処理され、また同時に第2命令サブセットIS2の各命令のデスティネーションオペランドは16ビット長として処理される。

【0053】一方、フラグレジスタの内容が"1"である場合には、上述とは逆に、第1命令サブセットIS1の各命令デスティネーションオペランドは第2ソースとして処理され、また同時に第2命令サブセットIS2の各命令のデスティネーションオペランドは第1ソースとして処理される。

【0054】従ってプログラマは、自身が作成するプログラムで扱われるオペランドのデスティネーションが主として第1ソースである場合にはフラグレジスタの内容が"0"に設定されるようにたとえばプログラムの最初にプログラミングする。そして、それ以降は、主として各命令コードが8ビットで構成される第1命令サブセットIS1を使用してプログラムを作成し、デスティネーションオペランドが第2ソースである命令が必要な場合のみ、第2命令サブセットIS2の命令を使用してプログラミングを行なう。

【0055】一方、プログラマは、プログラムで扱われるオペランドのデスティネーションが主として第2ソースである場合にはフラグレジスタの内容が"0"に設定されるようにたとえばプログラムの最初にプログラミングする。そして、それ以降は、主として各命令コードが8ビットで構成される第1命令サブセットIS1を使用してプログラムを作成し、デスティネーションオペランドが第1ソースである命令が必要な場合のみ、第2命令サブセットIS2の命令を使用してプログラミングを行なう。

【0056】このように、プログラム中で扱われる各命令のデスティネーションオペランドが主として第1ソースである場合にはフラグレジスタを"0"に、第2ソースである場合にはフラグレジスタを"1"にそれぞれセットすることにより、いずれの場合にも8ビット長の命令コードで構成される第1命令サブセットIS1を主として使用してプログラムを行なうことが可能になる。従って、第2命令サブセットIS2の16ビット長の命令コードは全く使用されないか、あるいは使用頻度が非常に低くなるので、プログラムサイズを小さくすることが可能になり、また命令の実行速度も高速化される。更に、フラグレジスタの内容をプログラムの途中で変更する必要もない。

【0057】なお、上述の本発明の命令セットのマイクロコンピュータによる実行方法の第2の実施例では、命令セットは基本命令セットとしての第1命令サブセットIS1と、この第1命令サブセットIS1の各命令の8ビットのコードに8ビットのを接頭コード"11111111"が付加された16ビットのコードを有する命令にて構成される第2命令サブセットIS2とで構成されているが、各命令セットに共通の下位コード長、換言すれば第1命令サブ

セットIS1のコード長は8ビットに限定されるものではない。また、接頭コードは“1111111”に限定されるものではなく、更に8ビットに限定されるものでもないが、基本の命令セットのコード長と同一ビット数とすることが望ましい。

【0058】

【発明の効果】以上に詳述したように本発明によれば、各命令の操作対象のオペランドサイズが複数の場合に、使用可能な命令の数を設定可能な命令の数に実質的に等しくすることを可能とし、またプログラミングの際のプログラムの負担を軽減し得る命令セット及びそのマイクロコンピュータにろう実行方法が実現される。

【0059】また、本発明によれば、各命令の演算結果の格納場所が複数の場合に、使用可能な命令の数を設定可能な命令の数に実質的に等しくすることを可能とし、またプログラミングの際のプログラムの負担を軽減し得る命令セット及びそのマイクロコンピュータによる実行方法が実現される。

【図面の簡単な説明】

【図1】 本発明の命令セットの各命令に割り付けられている命令コードを示す模式図である。

【図2】 本発明の命令セットのマイクロコンピュータによる実行方法の第1の実施例のフラグレジスタの内容と各命令セットにより操作されるオペランドサイズとの関係を示す模式図である。

【図3】 本発明の命令セット及びそのマイクロコンピュータによる実行方法の第2の実施例のフラグレジスタの内容と各命令セットによる演算結果の格納先との関係を示す模式図である。

【図4】 オペランドサイズが8ビットである場合に\*30

\*レジスタの内容とメモリの内容とをマイクロコンピュータにより実行される命令により加算してその結果を元のレジスタに格納する場合のデータの移動状態を示す模式図である。

【図5】 オペランドサイズが16ビットである場合にレジスタの内容とメモリの内容とをマイクロコンピュータにより実行される命令により加算してその結果を元のレジスタに格納する場合のデータの移動状態を示す模式図である。

10 【図6】 異なるサイズのオペランドを操作する場合の従来の命令セットの一例を示す模式図である。

【図7】 異なるサイズのオペランドを操作する場合の従来の命令セットの他の例を示す模式図である。

【図8】 レジスタの内容とメモリの内容とをマイクロコンピュータにより実行される命令により加算してその結果を元のレジスタに格納する場合のデータの移動状態を示す模式図である。

【図9】 レジスタの内容とメモリの内容とをマイクロコンピュータにより実行される命令により加算してその結果を元のメモリに格納する場合のデータの移動状態を示す模式図である。

【図10】 レジスタの内容とメモリの内容とを加算してその結果を元のレジスタに格納する場合の従来の命令セットの一例を示す模式図である。

【図11】 レジスタの内容とメモリの内容とを加算してその結果を元のメモリに格納する場合の従来の命令セットの一例を示す模式図である。

【符号の説明】

IS1 第1命令サブセット、IS2 第2命令サブセ

ット。

【図1】

二モニツク	IS1		IS2	
	第1命令サブセット 命令コード(2進表示)	第2命令サブセット 命令コード(2進表示)	第1命令サブセット 命令コード(2進表示)	第2命令サブセット 命令コード(2進表示)
ADD	00000000	11111111	00000000	11111111
SBB	00000001	11111111	00000001	11111111
AND	00000010	11111111	00000010	11111111
⋮	⋮	⋮	⋮	⋮

\*ただし、“11111111”は第1命令サブセットの命令コードとしては使用出来ない

【図2】

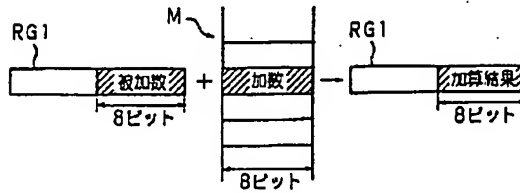
フラグレジスタの値	オペランドサイズ	
	第1命令サブセット	第2命令サブセット
0	8ビット	16ビット
1	16ビット	8ビット

【図3】

フラグレジスタの値	デスティネーション	
	第1命令サブセット	第2命令サブセット
0	第1ソース	第2ソース
1	第2ソース	第1ソース

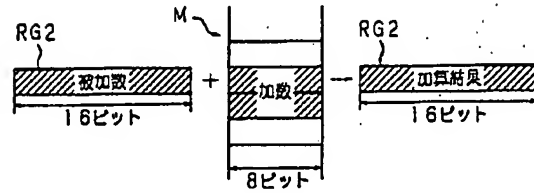
【図4】

オペランドサイズが8ビット長の場合



【図5】

オペランドサイズが16ビット長の場合

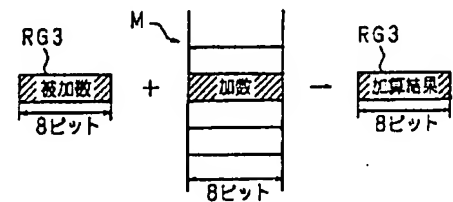


【図6】

ニーモニック	オペランドサイズ	命令コード(2進表示)
ADD.B	バイト( 8 ビット長)	00000000
ADD.W	ワード(16ビット長)	00000001
SUB.B	バイト( 8 ビット長)	00000010
SUB.W	ワード(16ビット長)	00000011
⋮	⋮	⋮

【図8】

デスティネーションが第1ソースの場合

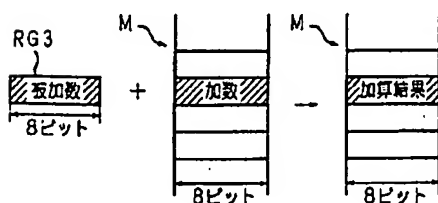


【図7】

ニーモニック	フラグレジスタ	オペランドサイズ	命令コード(2進表示)
ADD	1	バイト( 8 ビット長)	00000000
	0	ワード(16ビット長)	
SUB	1	バイト( 8 ビット長)	00000001
	0	ワード(16ビット長)	
⋮	⋮	⋮	⋮

【図9】

デスティネーションが第2ソースの場合



【図10】

ニーモニック	デスティネーション	命令コード(2進表示)
ADD.1	第1ソース(レジスタ)	00000000
ADD.2	第2ソース(メモリ)	00000001
SUB.1	第1ソース(レジスタ)	00000010
SUB.2	第2ソース(メモリ)	00000011
⋮	⋮	⋮

【図11】

ニーモニック	フラグレジスタ	デスティネーション	命令コード(2進表示)
ACD	1	第1ソース(レジスタ)	00000000
	0	第2ソース(メモリ)	
SUB	1	第1ソース(レジスタ)	00000001
	0	第2ソース(メモリ)	
⋮	⋮	⋮	⋮

## 【手続補正書】

【提出日】平成7年2月14日

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項4

【補正方法】変更

【補正内容】

【請求項4】 その内容を任意に設定可能なフラグレジスタを備え、複数のサイズのオペランドを操作する複数の命令にて構成された命令セットのマイクロコンピュータによる実行方法であって、

それぞれに割り付けられた異なるコードを有し、前記複数のサイズの内のいずれかのサイズのオペランドに対してそれぞれ所定の操作を実行する複数の命令にて構成された基本命令セットと、

それぞれに下位コードとして前記基本命令セットの各命令のコードが割り付けられると共に上位コードとして共通の接頭コードが付加されたコードを有し、付加されている接頭コードの内容に対応して前記複数のサイズの内のいずれかのサイズのオペランドに対して下位コードが同一の前記基本命令セットの命令と同一の操作を実行する複数の命令にてそれぞれが構成された複数の命令セットとを、

前記各命令セットが、前記フラグレジスタの内容と、接頭コードの有無、または付加されている接頭コードの内容とに応じてそれぞれ異なるサイズのオペランドを操作するようにしたことを特徴とする命令セットのマイクロコンピュータによる実行方法。

## 【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0014

【補正方法】変更

【補正内容】

【0014】次に、従来の技術の他の例としてたとえば、レジスタの内容とメモリの内容とにある演算、たとえば加算を行なってその結果を元のレジスタ、またはメ

モリに格納する場合のデータの移動状態を図8及び図9の模式図に示す。

## 【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0031

【補正方法】変更

【補正内容】

【0031】本発明の命令セットのマイクロコンピュータによる実行方法の第1の発明は、複数のサイズのオペランドを操作する複数の命令を実行する場合に、前述の命令セットの第1の発明の各命令セットをそれぞれが、その内容を任意に設定可能でマイクロコンピュータに備えられたフラグレジスタの内容と、接頭コードの有無、または付加されている接頭コードの内容とに応じてそれぞれ異なるサイズのオペランドを操作するように実行する。

## 【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0032

【補正方法】変更

【補正内容】

【0032】本発明の命令セットのマイクロコンピュータによる実行方法の第2の発明は、複数のサイズのオペランドを操作する複数の命令を実行する場合に、前述の命令セットの第2の発明の第1及び第2の命令セットを、その内容を任意に設定可能でマイクロコンピュータに備えられたフラグレジスタの内容が第1の状態にある場合は、第1の命令セットの各命令が第1のサイズのオペランドを操作し、第2の命令セットの各命令が第2のサイズのオペランドを操作するように、またフラグレジスタの内容が第2の状態にある場合は、第1の命令セットの各命令が第2のサイズのオペランドを操作し、第2の命令セットの各命令が第1のサイズのオペランドを操作するように実行する。

## 【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0033

【補正方法】変更

【補正内容】

【0033】本発明の命令セットのマイクロコンピュータによる実行方法の第3の発明は、第1のデータ格納場所に格納されたデータと第2のデータ格納場所に格納されたデータとの間で演算を実行して演算結果を第1または第2のデータ格納場所に格納する複数の命令を実行する場合に、前述の命令セットの第2の発明の第1及び第2の命令セットを、その内容を任意に設定可能でマイクロコンピュータに備えられたフラグレジスタの内容が第1の状態にある場合は、第1の命令セットの各命令は演算結果を第1の格納場所に格納し、第2の命令セットの各命令は演算結果を第2の格納場所に格納するように、またフラグレジスタの内容が第2の状態にある場合は、第1の命令セットの各命令は演算結果を第2の格納場所に格納し、第2の命令セットの各命令は演算結果を第1の格納場所に格納するように実行する。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0048

【補正方法】変更

【補正内容】

【0048】このように、プログラム中で扱われるオペランドのサイズが主として8ビットである場合にはフラグレジスタを“0”に、16ビットである場合にはフラグレジスタを“1”にそれぞれセットすることにより、いずれの場合にも8ビット長の命令コードで構成される第1命令サブセットIS1を主として使用してプログラムを作成することが可能になる。従って、第2命令サブセットIS2の16ビット長の命令コードは全く使用されないか、あるいは使用頻度が非常に低くなるので、プログラムサイズを小さくすることが可能になり、また命令の実行速度も高速化される。更に、フラグレジスタの内容をプログラムの途中で変更する必要もない。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0049

【補正方法】変更

【補正内容】

【0049】なお、上述の本発明の命令セットのマイクロコンピュータによる実行方法の第1の実施例では、命令セットは基本命令セットとしての第1命令サブセットIS1と、この第1命令サブセットIS1の各命令の8ビットのコードに8ビットの接頭コード“1111111”が付加された16ビットのコードを有する命令にて構成される第2命令サブセットIS2とで構成されているが、各命令セットに共通の下位コード長、換言すれば第1命令サブセットIS1のコード長は8ビットに限定されるものではない。

また、接頭コードは“1111111”に限定されるものではなく、更に8ビットに限定されるものでもないが、基本の命令セットのコード長と同一ビット数とすることが望ましい。

【手続補正8】

【補正対象書類名】明細書

【補正対象項目名】0052

【補正方法】変更

【補正内容】

【0052】ここでは、図1に示されている8ビットの命令コードからなる第1命令サブセットIS1と16ビットの命令コードからなる第2命令サブセットIS2とのマイクロコンピュータによる実行方法の第2の実施例として図3に示されているように、フラグレジスタの内容が“0”である場合には、第1命令サブセットIS1の各命令のデスティネーションオペランド、即ち演算結果の格納先が第1ソースとして処理され、また同時に第2命令サブセットIS2の各命令のデスティネーションオペランドは第2ソースとして処理される。

【手続補正9】

【補正対象書類名】明細書

【補正対象項目名】0055

【補正方法】変更

【補正内容】

【0055】一方、プログラマは、プログラムで扱われるオペランドのデスティネーションが主として第2ソースである場合にはフラグレジスタの内容が“1”に設定されるようにたとえばプログラムの最初にプログラミングする。そして、それ以降は、主として各命令コードが8ビットで構成される第1命令サブセットIS1を使用してプログラムを作成し、デスティネーションオペランドが第1ソースである命令が必要な場合のみ、第2命令サブセットIS2の命令を使用してプログラミングを行なう。

【手続補正10】

【補正対象書類名】明細書

【補正対象項目名】0057

【補正方法】変更

【補正内容】

【0057】なお、上述の本発明の命令セットのマイクロコンピュータによる実行方法の第2の実施例では、命令セットは基本命令セットとしての第1命令サブセットIS1と、この第1命令サブセットIS1の各命令の8ビットのコードに8ビットの接頭コード“1111111”が付加された16ビットのコードを有する命令にて構成される第2命令サブセットIS2とで構成されているが、各命令セットに共通の下位コード長、換言すれば第1命令サブセットIS1のコード長は8ビットに限定されるものではない。また、接頭コードは“1111111”に限定されるものではなく、更に8ビットに限定されるものでもないが、

基本の命令セットのコード長と同一ビット数とすることが望ましい。

【手続補正11】

【補正対象書類名】明細書

【補正対象項目名】0058

【補正方法】変更

【補正内容】

【0058】

【発明の効果】以上に詳述したように本発明によれば、各命令の操作対象のオペランドサイズが複数の場合に、使用可能な命令の数を設定可能な命令の数に実質的に等しくすることを可能とし、またプログラミングの際のプログラマの負担を軽減し得る命令セット及びそのマイクロコンピュータによる実行方法が実現される。